WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, December 23, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L11	L7 and ((scan or scanning) same method invocation)	0
☐	L10	L7 and scan$	9
☐	L9	L3 and (resource near3 exception)	1
☐	L8	L7 and (resource same exception)	4
☐	L7	L5 and (L1 or L2)	49
		DB=EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L6	L5	2
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L5	L4 and resource	754
☐	L4	L3 and exception	961
☐	L3	method invocation	2451
		DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ	
☐	L2	L1 or (712/233 |712/234 |712/235 |712/236 |712/237 |712/238 |712/239 |712/240 |712/241 |712/242 |712/243 |712/244).ccls.	4396
☐	L1	(717/131 |717/132 |717/133 |717/140 |717/141 |717/142 |717/143 |717/144 |717/146 |717/147 |717/148 |717/149 |717/150 |717/151 |717/152 |717/153 |717/154 |717/155 |717/156 |717/157).ccls.	2327

END OF SEARCH HISTORY

Web Results **11** - **20** of about **476** for **detecting scanning exception "method invocation"**. (**0.24** seconds)

Concurrent Programming in Java: State Dependence
... can be determined via reflection or **scanning** bytecodes ... from the change event, abort and rethrow the **exception**. ... a completed variable used for **detecting** run-time ...
www.awprofessional.com/articles/ article.asp?p=31539&seqNum=6 - 50k - Cached - Similar pages

[PDF] Manuscript Template for Intel Technology Journal.
File Format: PDF/Adobe Acrobat - View as HTML
... it may also need to detect relationships between ... Stack unwinding is required for **exception** handling, garbage ... 3] that assumes conservative **scanning** of the stack ...
www.intel.com/technology/itj/2003/ volume07issue01/art01_orp/vol7iss1_art01.pdf - Similar pages

GOODS programming interface for Java language
... JavaMOP propagates mutator attributes, iteratively **scanning** the methods ... To detect the moment of object modification ... final static int **EXCEPTION** = 8; // method is ...
www.ispras.ru/~knizhnik/goods/java/JavaAPI.htm - 53k - Cached - Similar pages

Understanding the DCOM Wire Protocol by Analyzing Network Data ...
... the MEOW structure is that when **scanning** through the ... service running on each machine to detect whether its ... for most method calls, with the **exception** of the ...
www.microsoft.com/msj/0398/dcom.aspx - 78k - Cached - Similar pages

The Hitch Hiker's Guide to the Smalltalk Compiler
... a collection of tokens, even after we scan in correctly ... during transformation to generate the **exception** signaling code ... The reason we wanted to detect code with ...
www.smalltalkchronicles.net/edition2-1/st_compiler.htm - 101k - Cached - Similar pages

[PDF] Practicing JUDO: Java Under Dynamic Optimizations
File Format: PDF/Adobe Acrobat
... For the ease of **detecting** common subexpressions in the IR construction ... our example we could add the following **method invocation** to the **exception** handler: e ...
dx.doi.org/10.1145/349299.349306 - Similar pages

[PDF] Removing Unnecessary Synchronization in Java
File Format: PDF/Adobe Acrobat - View as HTML
... state- ments of the form catch x as x = **Exception**.escapes . ... targets. Java's **method invocation** rules (sec. ... situation. 4.1 **Detecting** s-escaping objects ...
www.cs.ucsb.edu/labs/oocsb/papers/TRCS99-10.pdf - Similar pages

DyBASE - Object Oriented Database for languages with Dynamic Type ...
... setter method, it will not help to detect modification of ... APIs do not return False, but throw **exception** in case ... DyBASE performs cyclic **scanning** of bitmap pages ...
www.garret.ru/~knizhnik/dybase/doc/dybase.html - 74k - Cached - Similar pages

[PDF] Memory Management
File Format: PDF/Adobe Acrobat - View as HTML
... reasons these pointers must be found without **scanning** the heap ... It must detect this and switch to the new ... memory objects are short (with the **exception** of memory ...
www.opus.ub.uni-erlangen.de/opus/ volltexte/2004/37/pdf/08_Memory_Management.pdf - Similar pages

Guiding Principles

... collector must be capable of **scanning** these memory ... The virtual machine must detect illegal assignment attempts ... method occurs the thrown **exception** is discarded ...
rtj.org/public/src/javax/realtime/overview.html - 51k - Cached - Similar pages

IEEE Xplore® RELEASE 1.8

Welcome
United States Patent and Trademark ffice

Your search matched **3** of **1105713** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

method invocation<and>exception Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Semantic aspects of asynchronous RMI: the RMIX approach**
Kurzyniec, D.; Sunderam, V.;
Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International , 26-30 April 2004
Pages:157

[Abstract] [PDF Full-Text (1321 KB)] **IEEE CNF**

---

2 **Automated software engineering using concurrent class machines**
Grosu, R.; Liu, Y.A.; Smolka, S.; Stoller, S.D.; Jingyu Yan;
Automated Software Engineering, 2001. (ASE 2001). Proceedings. 16th Annua International Conference on , 26-29 Nov. 2001
Pages:297 - 304

[Abstract] [PDF Full-Text (974 KB)] **IEEE CNF**

---

3 **JGram: rapid development of multi-agent pipelines for real-world ta**
Sukthankar, R.; Brusseau, A.; Pelletier, R.; Stockton, R.;
Agent Systems and Applications, 1999 and Third International Symposium on Mobile Agents. Proceedings. First International Symposium on , 3-6 Oct. 1999
Pages:30 - 40

[Abstract] [PDF Full-Text (100 KB)] **IEEE CNF**

Terms used **method invocation scanning**

Found **86** of **148,162**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 86

Result page: **1** 2 3 4 5 next

Relevance scale

1 Protection traps and alternatives for memory management of an object-oriented language

Antony L. Hosking, J. Eliot B. Moss

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5

Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, citings, index terms

Many operating systems allow user programs to specify the protection level (inaccessible, read-only, read-write) of pages in their virtual memory address space, and to handle any protection violations that may occur. Such page-protection techniques have been exploited by several user-level algorithms for applications including generational garbage collection and persistent stores. Unfortunately, modern hardware has made efficient handling of page protection faults more difficult. Moreover, page- ...

2 Object fault handling for persistent programming languages: a performance evaluation

Antony L. Hosking, J. Eliot B. Moss

October 1993 **ACM SIGPLAN Notices , Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications**, Volume 28 Issue 10

Full text available: pdf(1.64 MB) Additional Information: full citation, references, citings, index terms

3 COOL: system support for distributed programming

Rodger Lea, Christian Jacquemot, Eric Pillevesse

September 1993 **Communications of the ACM**, Volume 36 Issue 9

Full text available: pdf(3.45 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: concurrency, concurrent object-oriented programming

4 Session: The COOL architecture and abstractions for object-oriented distributed operating systems

Rodger Lea, Christian Jacquemot

September 1992 **Proceedings of the 5th workshop on ACM SIGOPS European workshop: Models and paradigms for distributed systems structuring**

Full text available: pdf(594.76 KB) Additional Information: full citation, abstract, references, citings

Building distributed operating systems benefits from the micro-kernel approach by allowing better support for modularization. However, we believe that we need to take this support a step further. A more modular, or object oriented approach is needed if we wish to cross the barrier of complexity that is holding back distributed operating system development. The Chorus Object Oriented Layer (COOL) is a layer built above the Chorus micro-kernel designed to extend the micro-kernel abstractions with ...

5 Dynamic query evaluation plans

G. Graefe, K. Ward

June 1989 **ACM SIGMOD Record , Proceedings of the 1989 ACM SIGMOD international conference on Management of data**, Volume 18 Issue 2

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

In most database systems, a query embedded in a program written in a conventional programming language is optimized when the program is compiled. The query optimizer must make assumptions about the values of the program variables that appear as constants in the query, the resources that can be committed to query evaluation, and the data in the database. The optimality of the resulting query evaluation plan depends on the validity of these assumptions. If a query evaluation plan is used repe ...

6 A certifying compiler for Java

Christopher Colby, Peter Lee, George C. Necula, Fred Blau, Mark Plesko, Kenneth Cline

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5

Full text available: pdf(792.48 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents the initial results of a project to determine if the techniques of proof-carrying code and certifying compilers can be applied to programming languages of realistic size and complexity. The experiment shows that: (1) it is possible to implement a certifying native-code compiler for a large subset of the Java programming language; (2) the compiler is freely able to apply many standard local and global optimizations; and (3) the PCC bina ...

7 The Java programming language: Rethinking Java strings

Paolo Boldi, Sebastiano Vigna

June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java**

Full text available: pdf(55.14 KB) Additional Information: full citation, abstract

The Java string classes, String and StringBuffer, lie at the extremes of a spectrum (immutable, reference-based and mutable, content-based). Motivated by data-intensive text applications, we propose a new string class, MutableString, which tries to embody the best of both approaches.

8 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of

the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

9 Cost-driven vertical class partitioning for methods in object oriented databases

Chi-Wai Fung, Kamalakar Karlapalem, Qing Li

October 2003 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 12 Issue 3

Full text available: pdf(334.54 KB) Additional Information: full citation, abstract, index terms

Abstract.In object-oriented databases (OODBs), a method encapsulated in a class typically accesses a few, but not all the instance variables defined in the class. It may thus be preferable to vertically partition the class for reducing irrelevant data (instance variables) accessed by the methods. Our prior work has shown that vertical class partitioning can result in a substantial decrease in the total number of disk accesses incurred for executing a set of applications, but coming up with an op ...

Keywords: Affinity-based, Analytical cost model, Cost-driven, Hill-climbing heuristic algorithm, Method-induced, Object-oriented databases, Vertical class partitioning

10 Optimization techniques for queries with expensive methods

Joseph M. Hellerstein

June 1998 **ACM Transactions on Database Systems (TODS)**, Volume 23 Issue 2

Full text available: pdf(582.16 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Object-relational database management systems allow knowledgeable users to define new data types as well as new methods (operators) for the types. This flexibility produces an attendant complexity, which must be handled in new ways for an object-relational database management system to be efficient. In this article we study techniques for optimizing queries that contain time-consuming methods. The focus of traditional query optimizers has been on the choice of join methods and orders; selec ...

Keywords: expensive methods, extensibility, object-relational databases, predicate migration, predicate placement, query optimization

11 Profiling Java applications using code hotswapping and dynamic call graph revelation

Mikhail Dmitriev

January 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fourth international workshop on Software and performance**, Volume 29 Issue 1

Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references

Instrumentation-based profiling has many advantages and one serious disadvantage: usually high performance overhead. This overhead can be substantially reduced if only a small part of the target application (for example, one that has previously been identified as a performance bottleneck) is instrumented, while the rest of the application code continues to run at full speed. The value of such a profiling technology would increase further if the code could be instrumented and de-instrumented as m ...

12 Field analysis: getting useful and low-cost interprocedural information

Sanjay Ghemawat, Keith H. Randall, Daniel J. Scales

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5

Additional Information:

Full text available: pdf(686.96 KB) full citation, abstract, references, citings, index terms

We present a new limited form of interprocedural analysis called field analysis that can be used by a compiler to reduce the costs of modern language features such as object-oriented programming, automatic memory management, and run-time checks required for type safety. Unlike many previous interprocedural analyses, our analysis is cheap, and does not require access to the entire program. Field analysis exploits the declared access restrictions placed on fields in a modul ...

13 Distributed sensor network for real time tracking

Bryan Horling, Régis Vincent, Roger Mailler, Jiaying Shen, Raphen Becker, Kyle Rawlins, Victor Lesser

May 2001 **Proceedings of the fifth international conference on Autonomous agents**

Full text available: pdf(419.16 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we describe our solution to a real-time distributed resource allocation application involving distributed situation assessment. The hardware configuration consists of a set of reconfigurable sensors at fixed locations, each having local processing and low-bandwidth communication capabilities with other sensor nodes. The objective is to track objects moving in the environment in real-time as best as possible, given uncertainty and constraints on sensor loads, communication, p ...

14 Concrete syntax for objects: domain-specific language embedding and assimilation without restrictions

Martin Bravenboer, Eelco Visser

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available: pdf(379.91 KB) Additional Information: full citation, abstract, references, index terms

Application programmer's interfaces give access to domain knowledge encapsulated in class libraries without providing the appropriate notation for expressing domain composition. Since object-oriented languages are designed for extensibility and reuse, the language constructs are often sufficient for expressing domain abstractions at the semantic level. However, they do not provide the right abstractions at the syntactic level. In this paper we describe MetaBorg, a method for providing <i> ...

Keywords: MetaBorg, SDF, concrete object syntax, domain-specific languages, embedded languages, extensible syntax, meta programming, rewriting, stratego, syntax extension

15 How clean is the future of SOAP?

Conan C. Albrecht

February 2004 **Communications of the ACM**, Volume 47 Issue 2

Full text available: pdf(76.14 KB) html(16.40 KB) Additional Information: full citation, abstract, references, index terms

If developers are not wise with its application, SOAP may lose the ability to tunnel through firewalls---an ability that represents one of its primary advantages.

16 JRes: a resource accounting interface for Java

Grzegorz Czajkowski, Thorsten von Eicken

October 1998 **ACM SIGPLAN Notices , Pr ceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 33 Issue 10

Full text available: pdf(2.01 MB) Additional Information: full citation, abstract, references, citings, index terms

With the spread of the Internet the computing model on server systems is undergoing several important changes. Recent research ideas concerning dynamic operating system extensibility are finding their way into the commercial domain, resulting in designs of extensible databases and Web servers. In addition, both ordinary users and service providers must deal with untrusted downloadable executable code of unknown origin and intentions.Across the board, Java has emerged as the language of choice fo ...

Keywords: Java, extensible systems, resource management

17 An efficient implementation of SELF a dynamically-typed object-oriented language based on prototypes

C. Chambers, D. Ungar, E. Lee

September 1989 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 24 Issue 10

Full text available: pdf(2.41 MB) Additional Information: full citation, abstract, references, citings, index terms

We have developed and implemented techniques that double the performance of dynamically-typed object-oriented languages. Our SELF implementation runs twice as fast as the fastest Smalltalk implementation, despite SELF's lack of classes and explicit variables. To compensate for the absence of classes, our system uses implementation-level maps to transparently group objects cloned from the same prototype, providing data type information and eliminating the apparent ...

18 CyberCode: designing augmented reality environments with visual tags

Jun Rekimoto, Yuji Ayatsuka

April 2000 **Proceedings of DARE 2000 on Designing augmented reality environments**

Full text available: pdf(2.92 MB) Additional Information: full citation, abstract, references, citings, index terms

The CyberCode is a visual tagging system based on a 2D-barcode technology and provides several features not provided by other tagging systems. CyberCode tags can be recognized by the low-cost CMOS or CCD cameras found in more and more mobile devices, and it can also be used to determine the 3D position of the tagged object as well as its ID number. This paper describes examples of augmented reality applications based on CyberCode, and discusses some key characteristics of tagging technologies ...

Keywords: CyberCode, ID-aware interface, augmented reality, merging virtual and real

19 Design and specification of embedded systems in Java using successive, formal refinement

James Shin Young, Josh MacDonald, Michael Shilman, Abdallah Tabbara, Paul Hilfinger, A. Richard Newton

May 1998 **Proceedings of the 35th annual conference on Design automation - Volume 00**

Full text available: pdf(256.51 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Successive, formal refinement is a new approach for specificationof embedded systems using a general-purpose programming language.Systems are formally modeled as Abstractable SynchronousReactive systems, and Java is used as the design inputlanguage. A policy of use is applied to Java, in the form of languageusage restrictions and class-library

extensions, to ensureconsistency with the formal model. A process of incremental,user-guided program transformation is used to refine a Java programuntil ...

20 Practicing JUDO: Java under dynamic optimizations

Michał Cierniak, Guei-Yuan Lueh, James M. Stichnoth

May 2000 **ACM SIGPLAN Notices , Proceedings f the ACM SIGPLAN 2000 conference on Pr gramming language design and implementation**, Volume 35 Issue 5

Full text available: pdf(190.06 KB) Additional Information: full citation, abstract, references, citings, index terms

A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient implementation of Just-In-Time (JIT) compilation, exception handling, synchronization mechanism, and garbage collection (GC). These components are tightly coupled to achieve high performance. In this paper, we present some static anddynamic techniques implemented in the JIT compilation and exception handling of the Microprocessor Research Lab Virtual Machine (MRL VM), ...